ニューラルネットワークの形式的枠組みへの代数的な取り組み

著者	ラベルニア コントン ルイ マリー
雑誌名	東北大学電通談話会記録
巻	88
号	1
ページ	308-309
発行年	2019-07
URL	http://hdl.handle.net/10097/00126667

修士学位論文要約（平成31年３月）

ニューラルネットワークの形式的枠組みへの代数的な取り組み

ラベルニア　コントン　ルイ　マリー
指導教員：徳山 豪，　学位論文指導教員：全 眞嬉

An algebraic proposal on a formal framework of neural networks

Quentin Louis Marie LABERNIA
Supervisor: Takeshi TOKUYAMA,　Research Advisor: Jinhee CHUN

Neural networks form a very useful family of machine learning models. Training task recently showed a big improvement and the number of different models rose. Hence, given a specific dataset, the complex task turns out to be the designing of our neural network. Existing methods consider classical matrix and projections representation of neural networks, however, such point of view does not bring new perspective yet. We suggest a new point of view for classification task, where we link the common approach with a group theory and discrete mathematic based concepts. Instead of considering volumes as spaces, we flatten so that we can apply permutation operations to discrete linearly ordered sets. That is, we work with bead strings instead of multidimensional data. Such view brings a lot a new concepts applied to machine learning and perspectives to work on.

1. Background

The present work is dealing with machine learning, which recently expanded specifically thanks to deep learning advances. Neural networks are the main focus of the Master's thesis as they form a very interesting family of machine learning models. Typical deep learning model exhibits high accuracy on what are usually said to be hard problems: image, sound or video classification and segmentation, etc. On the other hand, designing neural networks became one of the main practical problems. Neural networks form a broad family of very different models, thus hyperparameters tuning is a crucial task to ensure the convergence of the model regarding to the provided training dataset. Our topic is the following: how can we efficiently guide the designing procedure of a neural network given a specific dataset? Hereafter, we rely on neural networks for classification.

2. State of the art

Even though hyperparameters have been studied, only a few results actually give concrete clues for neural network designing task. Neural network models consist of a lot of potential hyperparameters depending on the considered family, but those related to the network structure and training phase are the most common: for instance, the number of units in each layer, the connections or regularization and learning rate. The most recent advances rely on meta learning methods. Such algorithms take as input a dataset and the result is either a set of hyperparameters or even an already trained neural network, the hyperparameters of which were automatically chosen by the algorithm. Let us cite some of them: NAS and some derived works (such as ENAS or NASNet), AlphaX, AdaNet, ResFGB.

Most of the previous methods require a lot of computation power. Moreover, beside AdaNet and ResFGB algorithms, none of the previously given ones explicitly outputs a model whose properties are theoretically guaranteed. Example of such properties are generalization, learning stability or expressivity of the network. That is to say, blackboxed algorithms create blackboxed neural networks, and the main cause of this phenomenon is their internal complexity. Hence, we suggest a new way of considering neural networks.

3. Motivations

In order to derive new results and concepts, we want to look at the problem with new eyes. Namely, we aim at exploring a new face of neural networks. A major point is to simplify the ordinary matrix representation of neural networks and thus consider a discrete approach, rather the common continuous one. By doing so, we can study the internal structure in an easier way. Thus, we introduce a new view of classification and figure 1 illustrates the main intuitions.

Let us suppose that some notes are placed on your desk, and a label was written on each note. In order to classify such notes, we first exchange their position so that semantically similar notes become spatially close. Such process corresponds to a permutation and the movement of each notes creates a permutation graph. In order to finish our classification process, we need to be able to stack similar notes. In such case, the underlying graph is not a permutation graph anymore. This leads to the main contribution of this work, which is the definition of generalized permutation groups.

4. Contributions

We now recapitulate the different contributions of our work, the objective of which is to introduce a new framework. This framework aims at studying neural networks from a very perspective different from the common one, with simplified, approximated models.

We summarize our mathematical formalization and define so-called *generalized permutation groups*. Our group structure acts on linearly and finite orders. This represents the set of input instances, like a bead string instead of a volume. Such group is composed of two fundamental operations: transposition and fusion. Transpositions correspond to the classical view of permutation groups and its semantic is to relatively discriminate two given elements. Fusions allow group elements to gather instances -- two notes on the desk are superposed in such a way that they all follow the same movement from now on. For instance, based on X a finite and linearly ordered set, we create the group G using the elements:

$$\forall a, b \in X, a < b, f(a, b) \in G$$

with f being either tp (transposition) or fus (fusion) and $id \in G$ the identity element. Then, any composition of these functions is also in G, be definition of a group. We encode elements of X into $\overline{X}$ and let G act on this $\overline{X}$ which is an extended version of our bead string. In the case of image processing, $a \in G$ would be one image and its counterpart in $\overline{X}$ represents its state within the classification procedure. Formally, $\overline{x} \in \overline{X}$ has the following structure:

$$\overline{x} = (x, t, s), x \in X, t \in \mathbb{Z}, s \subset \mathrm{X}^2$$

The first component corresponds to the original element. The third one is a collection of pairs and record the history of all operations performed from the initial state in the classification process. Last, t just denotes the number of elements in s. Any classification can be modeled using this framework, as far as the input is discrete and finite.

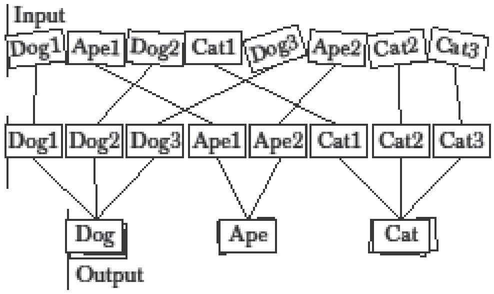

Figure 1　The classification process as permutations followed by some fusions. The first step corresponds to common permutation groups, while the second one is a key concept which makes it possible to define our so-called generalized permutation groups.

We also introduce an embedding method to consider any element of a generalized group G as a function of $X \to X$. We elaborate a construction which goes back and forth between X and its counterpart $\overline{X}$ using two specific functions $\uparrow$ and $\downarrow$. We prove the existence of such functions for any generalized group and finite and linearly ordered set X. This construction allows us to translate properties between the common context of study and our custom framework. As a complementary result to the previous embedding construction, we define a specific algorithm which allows to explicitly compute the algebraic expression in term of transpositions and fusions of any function $X \to X$. We express neural networks in our specific framework, so that their structure reflects its layered configuration. Using the previously mentioned algorithm, it is possible to convert any common neural networks into a combination of generalized permutation group elements.

Last, we refer to the symmetry of generalized group elements, which can be seen as a generalization of even and odd real functions. Based on this concept, we plan to investigate deeper into the structure of neural network since such symmetries naturally arises in neural networks expressed in our framework.

5. Conclusion

We define a new framework in which we can represents neural networks as generalized group elements. Lots of perspective come from this representation and the main one is to simplify the internal structure of deep neural networks and make them easier to design.